*Załącznik nr 1 do Uchwały nr \_\_\_\_\_\_\_ - 2018/2019 z dnia 25 marca 2019 r.*

*w sprawie wytycznych dla tworzenia i zmian programów studiów pierwszego stopnia, drugiego stopnia*

*oraz jednolitych studiów magisterskich rozpoczynających się od roku akademickiego 2019/2020.*

Opis **zajęć (sylabus)**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Nazwa zajęć: | | Programowanie obiektowe | | | | | | | **ECTS** | | **6** |
| Nazwa zajęć w j. angielskim: | | Object oriented programming | | | | | | | | | |
| Zajęcia dla kierunku studiów: | | **Informatyka** | | | | | | | | | |
|  | |  | | | | | | | | | |
| Język wykładowy: | | polski | | | | Poziom studiów: | | studia I stopnia | | | |
| Forma studiów: | 🞎 stacjonarne  🗷 niestacjonarne | Status zajęć: | 🗷 podstawowe  🞎 kierunkowe | 🗷 obowiązkowe  🞎 do wyboru | | Numer semestru: ……2….. | | 🞎 semestr zimowy 🗷 semestr letni | | | |
|  |  | Rok akademicki, od którego obowiązuje opis (rocznik): | | | | 2019/2020 | Numer katalogowy: |  | | | |
|  | | | | | | | | | | | |
| Koordynator zajęć: | |  | | | | | | | | | |
| Prowadzący zajęcia: | |  | | | | | | | | | |
| Jednostka realizująca: | |  | | | | | | | | | |
| Jednostka zlecająca: | |  | | | | | | | | | |
| Założenia, cele i opis zajęć: | | Zaznajomienie studentów z zasadami programowania obiektowego  Przekazanie studentom wiedzy na temat podstawowych zasad obiektowej analizy dziedziny i projektowania klas  Przekazanie studentom wiedzy o znaczeniu programowania obiektowego dla tworzenia dużych systemów  Nabycie przez studentów umiejętności posługiwania się środowiskiem programistycznym wspierającym programowanie obiektowe  Nabycie przez studentów umiejętności projektowania, implementowania, testowania i debugowania programów obiektowych a także właściwego wykorzystywania mechanizmów dziedziczenia, polimorfizmu i klas abstrakcyjnych. Opis tematów poruszanych podczas zajęć:Tematyka wykładów:  * Wprowadzenie do programowania obiektowego. Pojęcie klasy i obiektu. Składowe klasy - pola i metody. * Ochrona danych, hermetyzacja, specyfikacja dostępu do pól i metod, słowo kluczowe this. * Składowe statyczne. * Tworzenie, inicjalizacja i niszczenie obiektów. * Przeciążanie konstruktorów. Projektowanie klas. Obiektowe modelowanie dziedziny. Dziedziczenie. * Dziedziczenie a zawieranie. * Hierarchia klas, konstruktory a dziedziczenie. * Funkcje wirtualne i polimorfizm. * Klasy abstrakcyjne i interfejsy. Zastosowanie interfejsów. * Wyjątki. Przestrzenie nazw. * Przeciążanie operatorów, indeksatory, funkcje konwertujące. * Strumienie, praca z plikami, serializacja. * Kolekcje. * Typy ogólne. * Mechanizm refleksji..   Tematyka ćwiczeń laboratoryjnych:   * Informacje o środowisku Visual Studio i języku C#. Uwagi o Javie i środowisku NetBeans. * Tworzenie klas. Hermetyzacja. * Przykłady klas z .NET i ich zastosowania (Math, Convert, String, StringBuilder, Tablice i klasa Array). * Programowanie z wykorzystaniem dziedziczenia, funkcji wirtualnych oraz polimorfizmu. * Programowanie z wykorzystaniem interfejsów. * Programowanie z wykorzystaniem kolekcji. * Programowanie z zastosowaniem typów ogólnych. * Samodzielne tworzenie bibliotek obiektowych.. | | | | | | | | | |
| Formy dydaktyczne, liczba godzin: | | 1. wykład; liczba godzin ...18...; 2. ćwiczenia laboratoryjne; liczba godzin ...18...; | | | | | | | | | |
| Metody dydaktyczne: | | Wykład, dyskusja problemu, studium przypadków, pisanie programów, prezentacja i analiza kodów źródłowych, konsultacje | | | | | | | | | |
| Wymagania formalne  i założenia wstępne: | | Wstęp do Programowania, Matematyka Dyskretna. Konieczna podstawowa umiejętność programowania imperatywnego w językach strukturalnych oraz umiejętność projektowania prostych struktur danych | | | | | | | | | |
| Efekty uczenia się: | | Wiedza:  1 - Zna podstawowe zasady programowania obiektowego. Rozumie znaczenie programowania obiektowego dla tworzenia dużych systemów informatycznych  2 - Zna środowisko programistyczne wspierające pracę w wybranym obiektowym języku programowania, zna mechanizmy dziedziczenia i polimorfizmu, klasy abstrakcyjne oraz interfejsy  3 - zna przydatność paradygmatu obiektowego do rozwiązywania różnego typu problemów | | | Umiejętności:  1 - Ma umiejętność formułowania algorytmów i ich programowania z użyciem języka obiektowego  2 - potrafi dokonać obiektowej analizy dziedziny i zaprojektować strukturę klas dla danego zagadnienia i  3 - Potrafi ocenić przydatność środowiska programistycznego wspierającego pracę w wybranym obiektowym języku programowania. Potrafi wybrać i używać narzędzia do projektowania, implementowania, testowania i debugowania programów obiektowych | | | | | Kompetencje:  ……………………..  …………………….. | |
| Sposób weryfikacji efektów uczenia się: | | Zadania bieżące wykonywane podczas laboratorium. Sprawdzian praktyczny (kolokwia) podczas laboratorium  Egzamin pisemny praktyczny z pytaniami i zadaniami podobnymi do zadań z kolokwiów obejmujących całość materiału | | | | | | | | | |
| Forma dokumentacji osiągniętych efektów uczenia się: | | Zadania bieżące wykonywane podczas laboratorium (archiwum na Moodle).  Sprawdzian praktyczny (kolokwium) podczas laboratorium (archiwum na Moodle),  Egzamin pisemny z ocenami | | | | | | | | | |
| Elementy i wagi mające wpływ  na ocenę końcową: | | **Ćwiczenia laboratoryjne (wejściówki, zadania rozwiązywane na laboratorium) – 15%, sprawdzian praktyczny – 35%, egzamin – 50%.** | | | | | | | | | |
| Miejsce realizacji zajęć: | | Wykład -sala audytoryjna, ćwiczenia laboratoryjne – laboratorium komputerowe | | | | | | | | | |
| Literatura podstawowa i uzupełniająca:  1. Ian Graham: „Metody obiektowe w teorii i w praktyce”, WNT 2004.  2. Matt Weisfeld: Myślenie obiektowe w programowaniu– Helion 2009  3. Jesse Liberty, Brian MacDonald - C# 2005. Wprowadzenie – Helion 2007.  4. Bruce Eckel – „Thinking in C++” Edycja polska – Helion 2002  5. Bruce Eckel – „Thinking in Java” Edycja polska – Helion 2002.…  … | | | | | | | | | | | |
| UWAGI  Liczba punktów do zdobycia za bieżące ćwiczenia laboratoryjne: 30  Liczba punktów do zdobycia za sprawdzian praktyczny: 70  Liczba punktów do zdobycia za egzamin praktyczny pisemny: 100  Minimalna łączna liczba punktów konieczna do zaliczenia: 101 | | | | | | | | | | | |

Wskaźniki ilościowe charakteryzujące moduł/przedmiot:

|  |  |
| --- | --- |
| Szacunkowa sumaryczna liczba godzin pracy studenta (kontaktowych i pracy własnej) niezbędna dla osiągnięcia zakładanych dla zajęć efektów uczenia się - na tej podstawie należy wypełnić pole ECTS: | **150 h** |
| Łączna liczba punktów ECTS, którą student uzyskuje na zajęciach wymagających bezpośredniego udziału nauczycieli akademickich lub innych osób prowadzących zajęcia: | **2 ECTS** |

Tabela zgodności kierunkowych efektów uczenia się z efektami przedmiotu:

|  |  |  |  |
| --- | --- | --- | --- |
| kategoria efektu | Efekty uczenia się dla zajęć: | Odniesienie do efektów dla programu studiów dla kierunku | Oddziaływanie zajęć na efekt kierunkowy\*) |
| Wiedza 1 | Zna podstawowe zasady programowania obiektowego. Rozumie znaczenie programowania obiektowego dla tworzenia dużych systemów informatycznych | K\_W08 / P6S\_WG | 3 |
| Wiedza 2 | Zna środowisko programistyczne wspierające pracę w wybranym obiektowym języku programowania, zna mechanizmy dziedziczenia i polimorfizmu, klasy abstrakcyjne oraz interfejsy | K\_W10 / P6S\_WG | 2 |
| Wiedza 3 | zna przydatność paradygmatu obiektowego do rozwiązywania różnego typu problemów | K\_W17 / P6S\_WK | 1 |
| Umiejętności 1 | Ma umiejętność formułowania algorytmów i ich programowania z użyciem języka obiektowego | K\_U10 / P6S\_UW | 2 |
| Umiejętności 2 | potrafi dokonać obiektowej analizy dziedziny i zaprojektować strukturę klas dla danego zagadnienia | K\_U12 / P6S\_UW | 1 |
| Umiejętności 3 | Potrafi ocenić przydatność środowiska programistycznego wspierającego pracę w wybranym obiektowym języku programowania. Potrafi wybrać i używać narzędzia do projektowania, implementowania, testowania i debugowania programów obiektowych. | K\_U27 / P6S\_UW | 2 |
| Kompetencje - |  |  |  |
| Kompetencje - |  |  |  |

\*)

3 – zaawansowany i szczegółowy,

2 – znaczący,

1 – podstawowy,